In the Claims:

1. (Canceled).

2. (New) A computer system, comprising:

an integer pipeline having a plurality of integer stages;

a floating point pipeline having a plurality of floating point stages;

an instruction fetch stage shared by the integer and floating point pipelines;

a dedicated communications register coupled to the integer and floating point pipelines for enabling transfers between the integer and floating point pipelines; and

a memory;

wherein the integer pipeline can cause data in the memory to be transferred to the communications register.

3. (New) The computer system of claim 2, wherein the integer pipeline is capable of causing data in the communication register to be transferred to the memory.

4. (New) A computer system, comprising:

an integer pipeline having a plurality of integer stages;

a floating point pipeline having a plurality of floating point stages;

an instruction fetch stage shared by the integer and floating point pipelines; and

control circuitry for synchronizing the pipelines by synchronizing corresponding stages of the pipelines;

wherein at least one of the stages of one of the pipelines is coupled to the control circuitry, and wherein the control circuitry modifies a flow of operation of at least one of the stages of one of the pipelines, thereby synchronizing the pipelines;

wherein the control circuitry causes an instruction to be recirculated within a stage of one of the pipelines for the purpose of additional processing.

5. (New) A computer system, comprising:

an integer pipeline having a plurality of integer stages;

a floating point pipeline having a plurality of floating point stages;

an instruction fetch stage shared by the integer and floating point pipelines;

wherein the floating point pipeline includes data paths on which a result from an instruction can be passed to instructions in earlier stages of the floating point pipeline before that result can be stored in a register file.

6. (New) The computer system of claim 5, wherein results of floating point instruction can be sent to earlier stages of the integer pipeline and to be used by integer instructions.

7. (New) A computer system, comprising:

an integer pipeline having a plurality of integer stages;

a floating point pipeline having a plurality of floating point stages;

an instruction fetch stage shared by the integer and floating point pipelines;

wherein the integer pipeline includes data paths on which a result from an instruction can be passed to instructions in earlier stages of the integer pipeline before that result can be stored in a register file.

8. (New) A computer system, comprising:

an integer pipeline having a plurality of integer stages;

a floating point pipeline having a plurality of floating point stages;

an instruction fetch stage shared by the integer and floating point pipelines; and

control circuitry for synchronizing the pipelines by synchronizing corresponding stages of the pipelines;

wherein results produced by stages of the floating point pipeline are not compliant with industry standards regarding rounding or denormal numbers.

9. (New) The computer system of claim 8, wherein floating point results that have digits that should be rounded to a larger absolute value are always rounded to a smaller absolute value.

10. (New) The computer system of claim 9, wherein floating point operations that result in denormal numbers have their results replaced with zero.

11. (New) A pipeline having a plurality of floating point stages and control inputs through which external control circuitry controls the flow of instructions through the pipeline.

12. (New) The pipeline of claim 11, wherein one of the control inputs causes the pipeline to stall.

13. (New) The pipeline of claim 11, wherein the floating point stages include a decode stage, a first execution stage coupled to the decode stage and a write back stage coupled to the first execution stage.

14. (New) The pipeline of claim 11, wherein one of the control inputs causes the instruction in a decode stage to be replaced with a no operation instruction.

15. (New) The pipeline of claim 11, wherein one of the control inputs causes the instruction in an execution stage to be replaced with a no operation instruction.

16. (New) The pipeline of claim 11, wherein one of the control inputs causes the instruction in a write back stage to be replaced with a no operation instruction.

17. (New) The pipeline of claim 11, further comprising a plurality of additional execution stages coupled between a first execution stage and a write back stage.

18. (New) The pipeline of claim 11, wherein there is one control input related to each stage of the pipeline that will cause an instruction in that corresponding stage to be replaced with a no operation instruction.

19. (New) The pipeline of claim 11, wherein one of the control inputs will cause all instructions in all stages of the pipeline to be replaced with a no operation instruction.

20. (New) The pipeline of claim 11, wherein one of the control inputs causes an instruction in a decode stage to remain in the decode stage and a no operation instruction to be inserted in the execute stage after the previous instruction in the decode stage moves to a write back stage.

21. (New) A computer system comprising two pipelines, wherein each of the pipelines executes a unique set of instructions.

22. (New) The computer system of claim 21, wherein the pipelines each share a common instruction fetch stage that is coupled to a memory that stores instructions.

23. (New) The computer system of claim 22, wherein each of the pipelines includes a decode stage.

24. (New) The computer system of claim 23, wherein the fetch stage fetches instructions from a memory and passes them to the decode stage of each pipeline.

25. (New) The computer system of claim 24, wherein the decode stage of each pipeline decodes the instruction to see if the instruction should be executed in that pipeline.

26. (New) A computer system, comprising:

an integer pipeline having a plurality of integer stages;

a floating point pipeline having a plurality of floating point stages; and

an instruction fetch stage shared by the integer and floating point pipelines;

wherein a memory access address is calculated by the execution of the integer pipeline when the system is executing a floating point load or store.